101

SULLA NOZIONE DI "CALCOLABILE"

UNA RAPIDA LETTURA DELL'ARTICOLO DI TURING DEL 1936

Corso di Informatica Teorica - modulo 2

Prof. **Settimo Termini**

La questione

Lo scopo del corso, come già detto, è quello di presentare la *Teoria della Calcolabilità*, cercando di motivare l'idea che sia **ragionevole** definire in modo del tutto **generale** la nozione di funzione calcolabile.

(Questo, come vedremo, ci permetterà anche di capire alcuni dei motivi della pervasività dell'informatica nella nostra società)

Il problema, in realtà, è stato affrontato e risolto più di 70 anni fa, negli anni '30 del secolo XX. Lo discuteremo qui per vari motivi, **il primo è di tipo didattico**: ***Quando si presenta un nuovo argomento per quanto vecchio e già analizzato è ragionevole pensare che chi lo ascolta per la prima volta non conosce tutte le premesse a meno che queste non siano diventate parte integrante di una cultura condivisa e generale.***

La questione

La **seconda motivazione** trae origine dal fatto che - come già osservato - l'informatica oggi assolutamente pervasiva. Per capirne bene le sue potenzialità e i suoi pericoli è importante che una serie di sue caratteristiche generali siano analizzate a fondo. Il **terzo motivo** è che noi riteniamo che questi concetti generali siano molto sottili e profondi ma nello stesso tempo - fortunatamente - non richiedono - come spesso accade per altri concetti matematici o scientifici - forti presupposti di tipo formale. ***(Questo fatto renderebbe questi concetti ottimi e necessari candidati a diventare parte della cultura generale, cosa che oggi non è)***. L'ultimo punto infine riguarda una serie di problemi interessanti che tali nozioni coinvolgono e che sono connessi a problemi generali e a ricerche di oggi.

Un articolo particolare

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

by A. M. Turing

Proc. of the London Mathematical Society vol. 42 (1936-7), pp. 230-265

Un articolo particolare

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

by A. M. Turing

➡ *I numeri* ***computabili*** *possono essere brevemente descritti come i numeri reali le cui espressioni come numeri decimali sono calcolabili mediante risorse finite. Sebbene l'argomento di questo lavoro sia manifestamente quello dei numeri calcolabili, è egualmente facile definire e indagare le funzioni calcolabile di variabile intera o reale... I problemi di fondo sono tuttavia gli stessi in ciascun caso ed ho scelto in numeri calcolabili per una trattazione esplicita in quanto richiedono la tecnica meno pesante... Secondo la mia definizione un numero è calcolabile se il suo decimale può essere scritto da una macchina.*

Proc. of the London Mathematical Society vol. 42 (1936-7), pp. 230 - 265

L'ARTICOLO DEL 1936

Struttura del lavoro di Turing

1. Computing machines
2. Definitions
3. Examples of computing machines
4. Abbreviated tables
5. Enumeration of computable sequences
6. The universal computing machine
7. Detailed description of the universal machine
8. Application of the diagonal process
9. The extent of the computable numbers
10. Examples of large classes of number which are computable
11. Application to the Entscheidungsproblem

Struttura del lavoro di Turing

Ma vi è anche un'appendice:

Added 28 August, 1936.

APPENDIX.

Computability and effective calculability

The theorem that all effectively calculable (lambda-definable) sequences are computable and its converse are proved below in outline.

Struttura del lavoro di Turing

Definito questo aggeggio (la sua "Macchina"), a partire dalla sua "analisi del processo di calcolo", a seconda di come lo usiamo, dovremmo poter "definire" almeno:

- i numeri calcolabili

- le funzioni calcolabili

MA ANDIAMO AVANTI ADESSO A ZIG ZAG TRA ASPETTI STORICI, CONSIDERAZIONI GENERALI (E ANEDDOTI).

Quanti sono i “numeri calcolabili”?

Yet, Turing Machines- as these imaginary devices came to be called -cannot calculate every real number. The machines he designed have a finite number of operations, and by representing these operations with numbers, he is able to show that each machine can be uniquely described by a single integer called a Description Number. Turing Machines are thus enumerable. The computable numbers - the numbers that Turing Machines are capable of computing - must also be enumerable, but real numbers (we know from Cantor's proofs) are not enumerable. The computable numbers certainly include the algebraic numbers, and they also include such transcendental numbers as p greco and e, but because the computable numbers are enumerable, they simply cannot encompass all real numbers.

L'ultimo paragrafo

Diamo uno sguardo adesso all'ultimo paragrafo del lavoro di Turing, quello dedicato all'applicazione al "problema di decisione" della Teoria che ha sviluppato nei paragrafi precedenti.

The "ENTSCHEIDUNGSPROBLEM"

11. Application to the Entscheidungsproblem.

The results of §8 have some important applications. In particular, they can be used to show that the Hilbert Entscheidungsproblem can have no solution. For the present I shall confine myself to proving this particular theorem. For the formulation of this problem I must refer the reader to Hilbert and Ackermann's Grundziige der Theoretischen Logik (Berlin, 1931), chapter 3.

The "ENTSCHEIDUNGSPROBLEM"

(in traduzione italiana)

11. Applicazione al problema di decisione

I risultati del § 8 hanno alcune applicazioni importanti. in particolare essi possono essere usati per mostrare che il problema di decisione di Hilbert non può avere soluzione. al momento mi limiterò a dimostrare questo particolare teorema per la formulazione di questo problema devo rinviare il lettore al libro di Hilbert e Ackermann Grundzuge der Theoretischen Logik (Berlin, 1931), capitolo 3.

The "ENTSCHEIDUNGSPROBLEM"

Il libro, in realtà era stato pubblicato nel 1928. Il capitolo 3 è intitolato "Der engere Funktionenkalkul" o "Il calcolo funzionale ristretto" che con la terminologia di oggi chiameremmo: "Calcolo dei predicati del primo ordine". Hilbert e Ackermann nel libro scrivono:

"Il problema di decisione (Entscheidungsproblem) è risolto quando conosciamo una procedura con un numero finito di operazioni che determina la *validità* o *soddisfacibilità* di una data espressione ... Il problema di decisione si può considerare il problema principale della logica matematica."

The "ENTSCHEIDUNGSPROBLEM"

Osserviamo che Hilbert e Ackermann usano i termini validità e soddisfacibilità, termini che indicano che la metodologia per affrontare il problema che gli autori pensavano di seguire era di tipo semantico.

Turing invece usa un vocabolario differente per descrivere il problema di decisione:

"I propose, therefore, to show that there can be no general process for determining whether a given formula $\boldsymbol{U}$ of the functional calculus $\mathbf{K}$ is provable, i.e. that there can be no machine which, supplied with any one $\boldsymbol{U}$ of these formulae, will eventually say whether $\boldsymbol{U}$ is provable."

The "ENTSCHEIDUNGSPROBLEM"

Si vede subito quindi che Turing usando la parola *dimostrabile* invece di quelle di *validità* o *soddisfacibilità* mostra che vuole affrontare il problema di decisione con strumenti di tipo sintattico.

Possiamo ancora dire che, in base alle considerazioni svolte nei paragrafi 9 e 10, Turing può, a ragione, sostenere che se non si può progettare una macchina (del tipo di quelle che lui ha proposto) per implementare una procedura generale di decisione allora non può esistere neanche alcuna procedura generale di decisione che possa essere pensata da un essere umano

The "ENTSCHEIDUNGSPROBLEM"

Ma ci sono somiglianze e rapporti tra i risultati di Turing e quelli di Goedel? Già all'inizio del suo lavoro Turing aveva scritto che nel suo lavoro "conclusions are reached which are superficially similar to those of Goedel", ma Turing vuole essere più preciso:

The "ENTSCHEIDUNGSPROBLEM"

"It should perhaps be remarked that what I shall prove is quite different from the well-known results of Godel. Godel has shown that (in the formalism of Principia Mathematica) there are propositions ***U*** such that neither ***U*** nor **-*U*** is provable. As a consequence of this, it is shown that no proof of consistency of Principia Mathematica (or of **K**) can be given within that formalism. On the other hand, I shall show that there is no general method which tells whether a given formula ***U*** is provable in **K**, or, what comes to the same, whether the system consisting of **K** with **-*U*** adjoined as an extra axiom is consistent."

The "ENTSCHEIDUNGSPROBLEM"

e aggiunge:

"If the negation of what Goedel has shown had been proved, i.e. if, for each $\boldsymbol{U}$, either $\boldsymbol{U}$ or $-\boldsymbol{U}$ is provable, then we should have an immediate solution of the Entscheidungsproblem. For we can invent a machine K which will prove consecutively all provable formulae. Sooner or later K will reach either $\boldsymbol{U}$ or $-\boldsymbol{U}$. If it reaches $\boldsymbol{U}$, then we know that $\boldsymbol{U}$ is provable. If it reaches $-\boldsymbol{U}$, then, since **K** is consistent (Hilbert and Ackermann, p. 65), we know that $\boldsymbol{U}$ is not provable."

Una storia travagliata

E adesso un po' di storia:

Turing è stato spinto a "ideare" la sua macchina a partire dal problema della decisione come presentato dal suo professore Maxwell Neumann.

Molto probabilmente Turing aveva cominciato a lavorare al problema di decisione già nell'estate del 1935 e nel mese di aprile del 1935 aveva consegnato a Newman una bozza del suo lavoro.
Il lavoro era (ed è ancora) insolito come lavoro matematico, contenendo oltre a considerazioni di tipo generale proprio questo modello generale di "macchina" che poi lo renderà famoso.

L'ARTICOLO DEL 1936

Una storia travagliata

Mentre stava leggendo la bozza del lavoro di Turing, Newman ricevette un estratto di una noterella di Alonzo Church intitolata "A Note on the Entscheidungsproblem." che si basava su un suo lavoro precedente e si concludeva con la frase "***the Entscheidungsproblem is unsolvable***."

Questo fatto creava un problema grave per Turing

Normalmente un lavoro che presenta risultati già trovati (anche se da poco a anche se i risultati sono stati trovati in modo indipendente) non viene pubblicato. Ma Newman aveva capito che la via seguita da Turing era del tutto innovativa (oltre che diversa) da quella seguita da Church. E quindi gli suggerì di inviare lo stesso il lavoro alla London Mathematical Society. (La London Mathematical Society lo ricevette il 28 Maggio 1936.)

L'ARTICOLO DEL 1936

Una storia travagliata

Turing spiegò la situazione a sua madre in una lettera datata 29 maggio:

"Meanwhile a paper has appeared in America, written by Alonzo Church, doing the same things in a different way. Mr Newman and I decided however that the method is sufficiently different to warrant the publication of my paper too. Alonzo Church lives at Princeton so I have decided quite definitely about going there."

Una storia travagliata

Il 31 maggio Newman scrisse sia a Church sia al segretario della London Mathematical Society. A Church scrisse:

"An offprint which you kindly sent me recently of your paper in which you define 'calculable numbers', and show that the Entscheidungsproblem for Hilbert logic is insoluble, had a rather painful interest for a young man, A.M. Turing, here, who was just about to send in for publication a paper in which he had used a definition of 'Computable numbers' for the same purpose. His treatment - which consists in describing a machine which will grind out any computable sequence - is rather different from yours, but seems to be of great merit, and I think it of great importance that he should come and work with you next year if that is at all possible."

L'ARTICOLO DEL 1936

Una storia travagliata

Al segretario della London Mathematical Society scrisse:

"I think you know the history of Turing's paper on Computable numbers. just as it was reaching its final state an offprint arrived, from Alonzo Church of Princeton, of a paper anticipating Turing's results to a large extent. I hope it will nevertheless be possible to publish the paper. The methods are to a large extent different, and the result is so important that different treatments of it should be of interest. The main result of both Turing and Church is that the Entscheidungsproblem on which Hilbert's disciples have been working for a good many years - i.e., **the problem of finding a mechanical way of deciding whether a given row of symbols is the enunciation of a theorem provable from the Hilbert axioms - is insoluble in its general form**"

Una storia travagliata

Turing a questo punto aggiunge al suo lavoro un'appendice nella quale dimostra che il suo concetto di calcolabilità e la nozione di "calcolabilità effettiva" di Church sono equivalenti.

L'appendice è ricevuta dalla London Mathematical Society il 28 agosto 1936. E il lavoro è pubblicato nei Proceedings of the London Mathematical Society nei mesi di novembre e dicembre 1936.

Contemporaneamente, Alonzo Church in una recensione pubblicata sul Journal of Symbolic Logic nel mese di marzo 1937 scrive tra l'altro ***"a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a type of Turing machine"*** e questa è la prima volta in cui compare l'espressione *"Macchina di Turing"*.

Un articolo particolare

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

by A. M. Turing

RITORNIAMO A QUANTO GIA' LETTO:

➡ *"I numeri* ***computabili*** *possono essere brevemente descritti come i numeri reali le cui espressioni come numeri decimali sono calcolabili mediante risorse finite. Sebbene l'argomento di questo lavoro sia manifestamente quello dei numeri calcolabili, è egualmente facile definire e indagare le funzioni calcolabile di variabile intera o reale... I problemi di fondo sono tuttavia gli stessi in ciascun caso ed ho scelto in numeri calcolabili per una trattazione esplicita in quanto richiedono la tecnica meno pesante... Secondo la mia definizione un numero è calcolabile se il suo decimale può essere scritto da una macchina."*

E RIPORTIAMO ADESSO BREVI PEZZI DELL'ARTICOLO ORIGINALE

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach § 9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ which will be called "m-configurations". The machine is supplied with a "tape" (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol". At any moment there is just one square, say the r-th, bearing the symbol $\mathfrak{S}(r)$ which is "in the machine". We may call this square the "scanned square". The symbol on the scanned square may be called the "scanned

symbol". The "scanned symbol" is the only one of which the machine is, so to speak, "directly aware". However, by altering its m-configuration the machine can effectively remember some of the symbols which it has "seen" (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration q_n and the scanned symbol $\mathfrak{S}(r)$. This pair q_n, $\mathfrak{S}(r)$ will be called the "configuration": thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (*i.e.* bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down

will form the sequence of figures which is the decimal of the real number which is being computed. The others are just rough notes to "assist the memory". It will only be these rough notes which will be liable to erasure.

It is my contention that these operations include all those which are used in the computation of a number. The defence of this contention will be easier when the theory of the machines is familiar to the reader. In the next section I therefore proceed with the development of the theory and assume that it is understood what is meant by "machine", "tape", "scanned", etc.

OSSERVAZIONE:

IL “COMPUTER” PER TURING È UN UOMO CHE CALCOLA E LUI CERCA DI MODELLARLO ATTRAVERSO UNA MACCHINA

PASSIAMO IN RASSEGNA ANCORA QUALCHE BRANO:

I. [Type (*a*)]. This argument is only an elaboration of the ideas of § 1.

Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent†. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as

17 or 999999999999999 is normally treated as a single symbol. Similarly in any European language words are treated as single symbols (Chinese, however, attempts to have an enumerable infinity of symbols). The differences from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, cannot be observed at one glance. This is in accordance with experience. We cannot tell at a glance whether 999999999999999 and 999999999999999 are the same.

The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite.

The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

Let us imagine the operations performed by the computer to be split up into "simple operations" which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape. We know the state of the system if we know the sequence of symbols on the tape, which of these are observed by the computer (possibly with a special order), and the state of mind of the computer. We may suppose that in a simple operation not more than one symbol is altered. Any other changes

can be split up into simple changes of this kind. The situation in regard to the squares whose symbols may be altered in this way is the same as in regard to the observed squares. We may, therefore, without loss of generality, assume that the squares whose symbols are changed are always "observed" squares.

Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognisable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed squares does not exceed a certain fixed amount. Let us say that each of the new observed squares is within L squares of an immediately previously observed square.

In connection with "immediate recognisability", it may be thought

that there are other kinds of square which are immediately recognisable. In particular, squares marked by special symbols might be taken as immediately recognisable. Now if these squares are marked only by single symbols there can be only a finite number of them, and we should not upset our theory by adjoining these marked squares to the observed squares. If, on the other hand, they are marked by a sequence of symbols, we cannot regard the process of recognition as a simple process. This is a fundamental point and should be illustrated. In most mathematical papers the equations and theorems are numbered. Normally the numbers do not go beyond (say) 1000. It is, therefore, possible to recognise a theorem at a glance by its number. But if the paper was very long, we might reach Theorem 157767733443477; then, further on in the paper, we might find "... hence (applying Theorem 157767733443477) we have ...".

In order to make sure which was the relevant theorem we should have to compare the two numbers figure by figure, possibly ticking the figures off in pencil to make sure of their not being counted twice. If in spite of this it is still thought that there are other "immediately recognisable" squares, it does not upset my contention so long as these squares can be found by some process of which my type of machine is capable. This idea is developed in III below.

The simple operations must therefore include:

(*a*) Changes of the symbol on one of the observed squares.

(*b*) Changes of one of the squares observed to another square within L squares of one of the previously observed squares.

It may be that some of these changes necessarily involve a change of state of mind. The most general single operation must therefore be taken

to be one of the following:

(*A*) A possible change (*a*) of symbol together with a possible change of state of mind.

(*B*) A possible change (*b*) of observed squares, together with a possible change of state of mind.

The operation actually performed is determined, as has been suggested on p. 250, by the state of mind of the computer and the observed symbols. In particular, they determine the state of mind of the computer after the operation is carried out.

We may now construct a machine to do the work of this computer. To each state of mind of the computer corresponds an "*m*-configuration" of the machine. The machine scans *B* squares corresponding to the *B* squares observed by the computer. In any move the machine can change a symbol

Un articolo particolare

Quali sono le indicazioni che Turing ha tratto e che noi possiamo trarre da questa analisi?

- Numero finito di simboli
- numero finito di stati
- processi di base il più possibile elementari

VEDREMO SUCCESIVAMENTE COME DEFINIRE UNA “MACCHINA” A PARTIRE DA QUESTA ANALISI

La scommessa di questo articolo particolare:

sulla base di queste assunzioni estremamente elementari costruire una "teoria generale" da potere usare in tutti i contesti nei quali compare la nozione di "calcolabile": numeri, funzioni, manipolazione di simboli, etc.

SCOMMESSA RIUSCITA, come vedremo!